

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-111588

(43)Date of publication of application : 22.04.1994

(51)Int.Cl. G11C 16/06

(21)Application number : 04-252386

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 22.09.1992

(72)Inventor : ASANO HIDEO

(54) BATCH ERASING TYPE NONVOLATILE MEMORY

(57)Abstract:

PURPOSE: To provide a batch erasing type nonvolatile memory in which writing is not biased to a part of the memory.

CONSTITUTION: A processor includes plural accessible blocks to store a block ID expressing an address (RBA) designated by the processor at the time of writing, a rewriting code (RC) showing the number of writing times by the processor through the use of the same RBA and an erasing count (EC) showing the number of the erasing times of the pertinent block corresponding to each block. Writing is executed to a writable block of a smallest erasing count and when there is another block with the same block ID as the address designated by the processor at the time, its rewriting code is updated to make the rewriting code of the written block. In addition, the pertinent other block is erased and its erasing count is updated.

ブロック ID	RC	EC	EC0	ブロック ID
---------	----	----	-----	---------

LEGAL STATUS

[Date of request for examination] 22.09.1992

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2022761

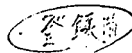
[Date of registration] 26.02.1996

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right] 31.05.2002

Copyright (C); 1998,2000 Japan Patent Office



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-111588

(43)公開日 平成6年(1994)4月22日

(51)Int.Cl.⁵
G11C 16/06

識別記号

庁内整理番号

FI

技術表示箇所

6741-5L

G11C 17/00

309 C

審査請求 有 請求項の数12(全 11 頁)

(21)出願番号 特願平4-252386

(22)出願日 平成4年(1992)9月22日

(71)出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72)発明者 浅野 秀夫

神奈川県藤沢市桐原町1番地 日本アイ・ビー・エム株式会社 藤沢事業所内

(74)代理人 弁理士 頓宮 孝一 (外4名)

(54)【発明の名称】 一括消去型不揮発性メモリ

(57)【要約】

【目的】 書込みがメモリの一部に片寄らないような一括消去型不揮発性メモリを提供すること。

【構成】 プロセッサがアクセス可能な複数のブロックを含み、各ブロックに対応して、書込み時にプロセッサが指定したアドレス(RBA)を表すブロックID、同じRBAを用いてプロセッサが何回書込みを行ったかを示す書換えコード(RC)及び当該ブロックの消去回数を示す消去カウント(EC)を記憶しておく。書込みは、消去カウントが最小の書込み可能なブロックに対して行い、そのときプロセッサが指定したアドレスと同じブロックIDを持った他のブロックがあれば、その書換えコードを更新して書込みを行ったブロックの書換えコードにすると共に、該他のブロックを消去しその消去カウントを更新する。

ブロックID	RC	EQ	ECC	データ(512バイト)
--------	----	----	-----	-------------

【特許請求の範囲】

【請求項1】 プロセッサがアクセス可能な複数のブロックを含む一括消去型不揮発性メモリであって、書き込み時に前記プロセッサが指定したアドレスを表すブロックID、該ブロックIDに対して前記プロセッサが何回書き込みを行ったかを示す書換えコード及び当該ブロックの消去回数を示す消去カウントを含む制御情報をブロック対応に記憶しておき、

前記プロセッサが所与のアドレスを指定して書き込みを要求したときは、前記消去カウントが最小の書き込み可能なブロックに対して書き込みを行うと共に、前記所与のアドレスを表すブロックIDを前記書き込み可能なブロックの制御情報に含め且つ該ブロックIDと同じブロックIDを持った他のブロックの書換えコードを更新して前記書き込み可能なブロックの制御情報とし、前記他のブロックを消去してその消去カウントを更新するようにしたことを特徴とする、一括消去型不揮発性メモリ。

【請求項2】 前記書き込み可能なブロックの制御情報に含まれる書換えコードはリセットされており、前記他のブロックがなければ、該書換えコードを最初の値に設定する、請求項1に記載の一括消去型不揮発性メモリ。

【請求項3】 前記書換えコード及び前記消去カウントの更新は、1だけ増分することによって行われる、請求項2に記載の一括消去型不揮発性メモリ。

【請求項4】 前記複数のブロックはそれぞれヘッダ部及びデータ部からなり、前記制御情報を該ヘッダ部に記憶する、請求項1に記載の一括消去型不揮発性メモリ。

【請求項5】 ブロックを消去したときは、そのヘッダ部に含まれる消去カウントを更新して該ブロックに書き戻す、請求項4に記載の一括消去型不揮発性メモリ。

【請求項6】 プロセッサからの相対ブロック・アドレスによってアクセスされる半導体ディスク装置であって、前記プロセッサに接続されたコントローラと、前記コントローラに接続され、前記プロセッサがアクセス可能な複数のブロックを含む一括消去型不揮発性メモリと、

前記コントローラに接続されたランダム・アクセス・メモリとを備え、

前記一括消去型不揮発性メモリの各ブロックはヘッダ部及びデータ部からなり、該ヘッダ部は、当該ブロックへの書き込み時に前記プロセッサが指定した相対ブロック・アドレスを表すブロックID、該ブロックIDに対して前記プロセッサが何回書き込みを行ったかを示す書換えコード及び当該ブロックの消去回数を示す消去カウントを含み、

前記コントローラは、前記プロセッサが所与の相対ブロック・アドレスを指定して書き込みを要求したときに、前記消去カウントが最小の書き込み可能なブロックのデータ部に前記プロセッサからのデータを書き込むと共に、そのヘッダ部に、前記所与の相対ブロック・アドレスによって

表されるブロックIDと、該ブロックIDと同じブロックIDを持った他のブロックの書換えコードを更新した値とを書込み、前記他のブロックを消去してそのヘッダ部の消去カウントを更新することを特徴とする半導体ディスク装置。

【請求項7】 前記書き込み可能なブロックの書換えコードはリセットされており、前記他のブロックがなければ、該書換えコードを最初の値に設定する、請求項6に記載の半導体ディスク装置。

【請求項8】 前記コントローラは、動作に先立って、前記一括消去型不揮発性メモリの各ブロックからヘッダ部の内容を順次に前記ランダム・アクセス・メモリに読み出して、各ブロックに対応するエントリを有するメモリ管理表を作成する、請求項7に記載の半導体ディスク装置。

【請求項9】 前記メモリ管理表の各エントリは、対応するブロックのヘッダ部の内容の他に、該対応するブロックの物理アドレスと、該対応するブロックが書き込み可能かどうかを示すフラグとを含む、請求項8に記載の半導体ディスク装置。

【請求項10】 前記コントローラは、前記メモリ管理表を作成したとき、ブロックIDが同じで且つ書換えコードが異なる2つのブロックがあれば、書換えコードが小さい方のブロックを消去してその消去カウントを更新すると共に、前記メモリ管理表における対応するエントリ中のフラグを書き込み可能状態にセットする、請求項9に記載の半導体ディスク装置。

【請求項11】 前記コントローラは、前記プロセッサからのコマンドに回答して前記メモリ管理表にアクセスし、読取りコマンドの場合は、該読取りコマンドの相対ブロック・アドレスを表すブロックIDを含むエントリ中の物理アドレスを用いて前記一括消去型不揮発性メモリの読取りを行い、書き込みコマンドの場合は、前記フラグが書き込み可能状態にセットされていて最小の消去カウントを含むエントリ中の物理アドレスを用いて前記一括消去型不揮発性メモリの書き込みを行う、請求項10に記載の半導体ディスク装置。

【請求項12】 前記コントローラは、前記メモリ管理表を作成するとき、読み取ったヘッダ部の書換えコードが消去によりリセットされていると、対応するエントリ中のフラグを書き込み可能状態にセットし、さもなければ書き込み不可状態にセットする、請求項11に記載の半導体ディスク装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、フラッシュEEPROMまたはフラッシュ・メモリと呼ばれる一括消去型不揮発性メモリに係り、特に、メモリ・ブロックの動的割り振りが可能な一括消去型不揮発性メモリ（以下、フラッシュ・メモリという）に係る。

【0002】

【従来の技術】ノートブック等の携帯可能なパーソナル・コンピュータの普及に伴って、コンピュータ・システムの小型軽量化、低消費電力化に対する要求が強くなってきている。半導体メモリを用いた外部記憶システム、いわゆる半導体ディスク装置は、磁気ディスク装置のように駆動系を持たないため、消費電力が低く、高速動作が可能である。また、小さなメモリ・モジュールで構成されるため、磁気ディスク装置に比べて小型で軽く、形状に自由度が大きく、カード化も容易である。

【0003】しかし、従来の半導体メモリにはコスト、容量、電池バックアップなどの点でまだ問題が多い。メモリとしてSRAMを使うと電池によるバックアップの時間は長くなるが、コストが高く、容量も小さくなってしまふ。コスト、容量に優れたDRAMでは、スタンバイ時の消費電力が大きく、バックアップの時間が一週間程度に限られてしまふ。電池系の事故によるデータ消失の危険もある。EEPROMは電池を必要としないが、コストが高すぎる。

【0004】これらの問題を解決するメモリとして一括消去型のフラッシュ・メモリが開発されている。DRAMと同じくトランジスタ1つで記憶素子が構成され、高密度化が可能で、将来の市場次第でDRAMと同等かそれ以下のビット単価（低コスト、大容量）になることが期待されている。記憶素子是不揮発性であり、電池バックアップの必要はない。消去は一般にチップ単位又はそれよりも小さなブロック単位で行われる。Richard D. Pashley 外の "Flash memories: the best of two worlds"、IEEE SPECTRUM 1989年12月、30~33頁は、このようなフラッシュ・メモリの概要を紹介している。パフォーマンスの点では、チップ消去型よりブロック消去型の方が優れている。

【0005】ブロック消去型のフラッシュ・メモリを半導体ディスク装置に用いる場合は、ブロックの大きさを磁気ディスク装置のアクセス単位であるセクタに等しくしておくと、メモリ管理に都合がよい。例えば、ヨーロッパ公開特許出願第392895号はセクタ消去型のフラッシュEEPROMシステムを開示している。このシステムは、消去単位であるセクタ毎にラッチを設けておき、消去したいセクタに対応するラッチをセットすることによって、任意の複数のセクタを同時に消去できるようにしている。複数セクタ分の容量（例えば4Kバイト）を持ったブロックを消去単位にしたフラッシュ・メモリも知られており、これはセクタ消去型と区別するためにクラスタ消去型と呼ばれることがある。

【0006】しかし、フラッシュ・メモリにはSRAMやDRAMにはない制限がある。まず、メモリ・ビットのプログラミングは一方通行で、0から1又は1から0へしか変えることができない。従って、既に書込まれている記憶位置に新たなデータを書込む場合は、その記憶

位置を含むブロックを一括消去によって全0又は全1に設定した後に書込みを行う必要がある。消去及び書込みには、通常、数十ミリ秒から数秒の時間がかかる。また、フラッシュ・メモリは消去及び書込みによって劣化し、現在のところ、数万回から数十万回の消去及び書込みで使用限度に達してしまう。

【0007】

【発明が解決しようとする課題】このようなフラッシュ・メモリを半導体ディスク装置に用いた場合、書込みがメモリの一部に片寄ることが問題になる。例えば、DOSベースのパーソナル・コンピュータ・システムでは、ファイル・アロケーション・テーブル（FAT）の書換えがしばしば行われる。ところが、FATのアドレスは固定されているため、フラッシュ・メモリを用いた場合には、FATの書換えの度にそれを記憶しているブロックの消去及び書込みを行う必要があり、その度に数十ミリ秒から数秒の時間がかかってしまふ。また、このように書込み及び消去がメモリの一部のブロックに片寄ると、そのブロックが他のブロックに比べて早く使用限度に達してしまひ、他のブロックがまだ使用可能であるにもかかわらず、メモリを交換しなければならなくなる。使用限度に達したブロックを無効化して、代替ブロックを使用するようにすれば、メモリの早期交換は避けられるが、書込みが集中するのが代替ブロックに変わったただけであり、根本的な解決にはなっていない。

【0008】更に、前述のようにフラッシュ・メモリの消去及び書込みには数十ミリ秒から数秒の時間がかかるため、その最中にパワーオフがあると、要求された消去又は書込みを完了仕切れないことがある。その場合は、消去又は書込みが行われていたブロックの内容は予測不能であり、従ってフラッシュ・メモリには異常パワーオフ対策も必要である。

【0009】従って本発明の目的は、書込みがメモリの一部に片寄らないようなフラッシュ・メモリを提供することにある。

【0010】本発明の他の目的は、消去又は書込み中にパワーオフがあってもその後の動作に支障がないようなフラッシュ・メモリを提供することにある。

【0011】

【課題を解決するための手段】本発明に従うフラッシュ・メモリは、プロセッサがアクセス可能な複数のブロックを含む。書込みがメモリの一部に片寄らないようにするため、本発明では、プロセッサから書込みコマンドがくると、プロセッサが指定したアドレスとは無関係に、各ブロックの消去回数に基づいて書込むべきブロックを動的に割り振る。消去回数は、各ブロックの制御情報中の消去カウントにより示される。制御情報は、消去カウントの他に、そのブロックに対して書込みを行ったときにプロセッサが指定したアドレスを表すブロックIDを含み、更にこの同じブロックIDに対してプロセッサが

何回書込みを行ったかを示す書換えコードを含む。書込みは、消去カウントが最小の書換え可能なブロックに対して行われ、そのとき制御情報も更新される。書換えコードがリセットされているブロックは書込み可能である。また、書込みを行うとき、そのブロックIDと同じブロックIDを含む古いブロックがあれば、その書換えコードを更新して新たな書込みブロックの制御情報とし、古いブロックは消去して、その消去カウントを更新する。

【0012】後述の実施例においては、各ブロックはヘッダ部及びデータ部からなり、制御情報はヘッダ部に記憶される。ブロックを消去したときは、消去カウントを1だけ増分してブロックのヘッダ部に書き戻す。そのとき書換えコードは消去された状態、すなわちリセットされたままであり、このブロックが書込み可能であることを示す。各ブロックのヘッダ部の内容は、動作に先立ってコントローラにより読み出される。コントローラはそれを用いて自身のメモリ（ランダム・アクセス・メモリ）にメモリ管理表を作成する。この表は、フラッシュ・メモリの各ブロックに対応するエントリを含む。各エントリは、対応するブロックの物理アドレスと、ヘッダ部から読み出した情報すなわちブロックID、書換えコード及び消去カウントと、このブロックが書込み可能かどうかを示すフラグとを含む。コントローラは、メモリ管理表を作成したとき、ブロックIDが同じで書換えコードが異なる2つのブロックがあれば（この状態は、異常パワーオフのため古いブロックを消去できなかったときに生じ得る）、書換えコードが小さい方のブロック、すなわち古い方のブロックを消去し、その消去カウントを更新する。コントローラは、このメモリ管理表を用いてプロセッサからのコマンドを実行し、必要に応じて関連するエントリを更新する。

【0013】

【実施例】本発明のフラッシュ・メモリを半導体ディスク装置として組み込んだコンピュータ・システムの一例を図1に示す。CPU10はシステム・バス13を介して、主記憶装置15、バス制御装置16及びオプションの数値計算用コプロセッサ14と通信する。CPU10及び関連する周辺装置の間の通信はバス制御装置16を介して行われる。そのため、バス制御装置16はファミリー・バス18によって周辺装置に接続されている。周辺装置としては、本発明に従うフラッシュ・メモリ製の半導体ディスク装置（SSF）20が接続され、さらに、通信装置21、フロッピー・ディスク・ドライブ（FDD）22、光ファイル装置23、表示装置24もファミリー・バス18に接続されている。勿論、他の周辺装置も接続可能である。このようなコンピュータ・システムの一例はIBM PS/2である。

【0014】直接メモリ・アクセス制御装置（DMAC）12は、これらの周辺装置の全部又は選択された何

台かによるメモリ・アクセスを可能にすべく設けられる。そのため、ファミリー・バス18は、少なくともその一部がDMAC12に分岐接続される。図には示していないが、DMAが可能な各周辺装置にはアービトレーション回路が設けられ、アービトレーション・レベル（優先順位）を割り当てられる。DMAC12の側には、DMAを同時に要求している複数の周辺装置の間で調停作業を行って、どの周辺装置がDMAを許可されたかをDMAC12に知らせる中央アービトレーション制御回路11が設けられる。DMAC12及び中央アービトレーション制御回路11によるDMA制御の詳細は米国特許第4901234号明細書に記載されている。

【0015】CPU10はSSF20をハード・ディスク装置として扱う。従って、SSF20をアクセスするときは、ヘッド番号、シリンダ番号及びセクタ番号から成るいわゆる相対ブロック・アドレス（RBA）がSSF20に送られる。ただし、後で述べるように、本発明においては、CPU10から供給されるRBAと、SSF20の実際にアクセスされるブロックのアドレス（物理アドレス）との間の関係は固定されておらず、書込みの度に変化するので、それらの対応関係を明らかにする手段（メモリ管理表）が設けられる。CPU10からのRBAはいわば論理アドレスである。

【0016】図2に、SSF20の概略的な構成を示す。このSSF20は、ファミリー・バス18に接続された制御装置30と、内部バス31を介してこの制御装置30に接続されたランダム・アクセス・メモリ（RAM）32、バス制御部33及びフラッシュ・メモリ34で構成される。RAM32は、後述のメモリ管理表を記憶する領域35及びバッファ領域36を含む。バス制御部33は、内部バス31と、フラッシュ・メモリ34に接続されたメモリ・バス37とを相互接続するための周知のレシーバ／ドライバ構成を有する。本実施例では、バス制御部33はバス31及び37間で2バイトの並列転送を行うようになっているが、勿論本発明はこのような並列転送に限定されるものではない。また、本発明に関する限り、フラッシュ・メモリ34の消去型はいずれでも構わないが、以下では説明の便宜上、フラッシュ・メモリ34をセクタ消去型とする。従って、1ブロック＝1セクタである。

【0017】フラッシュ・メモリ34の各ブロック（セクタ）の構成を図3に示す。図示のように、各ブロックは512バイトのセクタ・データを記憶するデータ部の他に、ヘッダ部を含んでいる。ヘッダ部は、このブロックに書込みを行ったときにCPU10が指定した相対ブロック・アドレス（RBA）によって表されるブロックID、同じRBAのブロックが何回書換えられたかを示す書換えコード（RC）、この物理ブロックの消去回数を示す消去カウント（EC）及びブロック全体のエラーを検査し訂正するためのエラー訂正符号（ECC）を含

む。工場出荷時には、ブロックID及びRCはいずれも0に初期設定されるが、ECの初期設定値は出荷前のテストで当該ブロックを消去した回数を表す数値nである。RC及びECのビット数は、フラッシュ・メモリの公称の消去/書込み上限回数よりも大きい数表せるように選んでおいた方がよい。本実施例では、RC及びECはいずれも4バイトである。ブロックID、RC及びECはそれぞれエラーを検出するためのパリティ・ビットをバイト毎に含んでいる。これは次に述べるメモリ管理表を作成するとき必要である。

【0018】図2に示した制御装置30はパワーオン時に各ブロックのヘッダ情報を読み、図4に示すようなメモリ管理表をRAM32の領域35に作成する。メモリ管理表の各エントリはフラッシュ・メモリ34のブロックと1対1に対応している。従って、フラッシュ・メモリ34がn個のブロックで構成されていると、メモリ管理表は最大n個のエントリを含む。制御装置30は、欠陥のある無効なブロックについてはエントリを作成しない。各エントリは、対応するブロックの物理アドレスと、このブロックのヘッダ部に含まれている相対ブロック・アドレス（ヘッド番号/シリンダ番号/セクタ番号）、書換えコード（RC）及び消去カウント（EC）と、書込み可能フラグ（OW）とを含む。書込み可能フラグは、対応するブロックが空で書込みが可能である（OW=0）か、又は有効なデータが存在するため書込みが不可である（OW=1）ことを示す。メモリ管理表を作成するとき、ヘッダ部に含まれる書換えコードが0であれば、対応するエントリ中の書込み可能フラグは0に設定され、さもなければ1に設定される。

【0019】次に、図5のフローチャートを参照しながら、CPU10からコマンドを受け取ったときのSSF20の動作について説明する。SSF20の制御装置30はステップ100でコマンドを受け取ると、次のステップ101でそれをデコードする。読取りコマンドの場合はステップ102に進み、書込みコマンドの場合はステップ104に進む。まず書込みコマンドについて説明する。

【0020】ステップ101で書込みコマンドが検出されると、制御装置30はCPU10からのデータをバッファ36に書込み、ステップ104に進む。ステップ104では、メモリ管理表35をアクセスすることにより、OWが1で且つこの書込みコマンドのRBAと同じRBAのブロックがあるかどうかを調べる。もしあればステップ105に進んで、同じRBAを含んでいたエントリ中の物理アドレス、RC及びECを取り出して保管する。物理アドレスを保管するのは、後でこの物理アドレスによって指定されるブロックを消去するためである。ステップ105でブロックを消去してしまうと、異常パワーオフがあったときにRCを回復できないことがあるので好ましくない。この点については後で説明す

る。

【0021】ステップ105が終了するか、またはステップ104で同じRBAのブロックが見つからなければ、次のステップ106に進む。ステップ106では、メモリ管理表35をアクセスすることにより、OWが0で且つECが最小のブロックを探す。そのようなブロックすなわち書込みが可能で且つこれまでの消去回数が最小のブロックが見つかったら、ステップ107へ進んで、そのブロックへの書込み及びメモリ管理表35の更新を行う。

【0022】ブロック書込みは、メモリ管理表35の対応するエントリにある物理アドレスを用いて行う。ブロックに書込まれるのはCPU10からのデータ（バッファ36にある）及びECを除く前述のヘッダ情報である。後で説明するように、ECは既に更新された値が書込まれている。制御装置30は、ステップ105で保管しておいたRCを1だけ増分して、新たなヘッダ情報とする。ステップ105が実行されなかった場合は、CPU10が指定したRBAの初めての書込みであるから、RCは1にセットされる。またヘッダ情報として書込むべきブロックIDは、ステップ101で検出した書込みコマンドのRBAをそのまま使用する。制御装置30は、このようにして更新されたRC及びブロックIDのパリティ・ビットを生成し、それをRC及びブロックIDに付加した後、バッファ36から読み出したデータと共に、指定されたブロックに書込む。ECは、更新されたRC及びブロックIDと、メモリ管理表の対応するエントリ中のECと、バッファ36から読み出したデータとを用いて生成され、ヘッダ情報として同じブロックに書込まれる。メモリ管理表35の対応するエントリもこれらの新しいRBA及びRCを用いて更新され、且つそのOWは1にセットされる。

【0023】ステップ106で複数のブロックが見つかった場合は、そのうちの1つ（例えば物理アドレスがもっとも小さいブロック）に書込まれる。一般に、フラッシュ・メモリは予備のブロックを含んでいるので、OWが0である書込み可能なブロックが見つからないということはない。

【0024】ステップ107でのブロック書込みが終了すると、ステップ105で保管しておいた物理アドレスによって指定されるブロックを消去し、同じくステップ105で保管しておいたECを1だけ増分してパリティ・ビットを付加した後この消去済みのブロックのヘッダ部に書込み、更にメモリ管理表35の対応するエントリを $EC = EC + 1$ 及び $OW = 0$ となるように更新する（ステップ108）。

【0025】ステップ108が終了すると、制御装置30はCPU10からの次のコマンドを待つ。制御装置30はこのときステップ106を実行することによって、次に書込むべきブロックを予め選択しておいてもよい。

そうすれば、CPU 10から新たな書込みコマンドがきたときに、ステップ106が不要になり、書込みサイクルを短縮することができる。

【0026】読取りコマンドの場合は、ステップ101から102に進み、CPU 10からのRBA（ヘッド番号／シリンダ番号／セクタ番号）を用いて図4のメモリ管理表35をアクセスすることにより、読み取るべきブロックの物理アドレスを得る。次のステップ103では、この物理アドレスをフラッシュ・メモリ32へ供給することにより、指定されたブロックを読取り、そのヘッダ部のECCを用いてエラーを検査する。ステップ109で訂正不能エラーが見つからなければ、読み取ったブロックに含まれる512バイトのデータを、RAM32のバッファ領域35を介してCPU 10の方へ転送する。バッファ領域を介する転送は、ハード・ディスク装置で従来行われていたものと同じである。

【0027】訂正不能エラーがあると、ステップ110に進む。読み取ったブロックに訂正不能エラーが生じているということは、そのブロックに対する以前の書込み中にパワーオフがあって、データ内容が乱された可能性があることを示す。ステップ110では、訂正不能エラーが生じたことをCPU 10に知らせ、そのブロックを消去する。

【0028】また、訂正不能エラーが検出された場合、後で説明する図6のシーケンスによって作成されたメモリ管理表の対応するエントリ中のECに、通常のパリティ検査では検出できないエラーが生じている可能性があるため、ステップ110で消去したブロックのこれまでの消去回数を推定する必要がある。本実施例では、その推定値として隣接ブロック（例えば、物理アドレスが1つ前又は1つ後ろのブロック）のECを使用する。このECはメモリ管理表から得ることができる。信頼性をより重視するのであれば、フラッシュ・メモリ34における実際の隣接ブロックから読み出したECを用いればよい。一般に、隣接するブロック間では、消去回数に殆ど差はないので、ステップ110で消去したブロックのECとして隣接ブロックのECを使用しても実用上問題はない。計算が必要になるが、隣接する幾つかのブロックのECの平均値を使用することも可能である。次のステップ111は、このようにして得たECを1だけ増分した後、消去済みのブロックに書込む。最後に、ステップ112でメモリ管理表の対応するエントリを更新する。すなわち、新しいECを書込み、OWを0にセットする。

【0029】なおステップ101では、読取り及び書込みの他に、クリア、状況読取り等のコマンドもデコードされるが、それらは本発明とは関係がないので、ここでは省略する。

【0030】次に、異常パワーオフ対策を明らかにするため、パワーオン後のメモリ管理表の作成について図6

を参照しながらもう少し詳しく説明する。前述のように、制御装置30はパワーオン時にフラッシュ・メモリ34を構成するn個のブロックのヘッダを順次読み出すことにより、図4に示すようなメモリ管理表を作成する。そのため、ステップ200でパワーオンがあると、次のステップ201でブロック・アドレスBAを0に初期設定し、ステップ202でBA=0のヘッダを読み取る。データ部の読取りは不要である。次のステップ203では、パリティ検査により、ヘッダにエラーが生じているかどうかを調べる。エラーが生じていなければステップ204に進み、生じているとステップ205に進む。

【0031】ステップ204では、ステップ202で読み取ったヘッダ情報を用いてメモリ管理表のエントリを作成する。まず物理アドレスは、ヘッダ読取り時のアドレスBAを使用する。ブロックID、RC及びECはヘッダの内容をそのまま使用する。但しブロックIDに関しては、そのヘッド番号、シリンダ番号及びセクタ番号の各ビット数は余裕をもって設定されており、普通は上位数ビットは0になっているので、エントリ作成に当たっては、これらの上位ビットを落しても差し支えない。そうすると各エントリのビット数を節約できる。最後の書込み可能フラグOWは、読み取ったRCが0であれば0にセットされ、さもなければ1にセットされる。RCが0ということは、それを含むブロックが消去されていること、従って書込みが可能であることを表す。なお図4には示していないが、エラーの検出訂正のため、各エントリにECCを付けておくのが望ましい。

【0032】エントリの作成が終了すると、ステップ206でブロック・アドレスBAを1だけ増分し、ステップ207でn個全てのブロックのヘッダの読取りが終了したかどうかを調べる。まだ終わっていない場合は、ヘッダ読取りステップ202に戻り、ステップ202～207を繰り返す。

【0033】ステップ202で読み取ったヘッダにエラーがあると、ステップ203から205へ分岐する。ステップ205では、ブロック全体を読み取ってECCを適用することにより、そのエラーが訂正可能かどうかを調べる。訂正可能であれば、ステップ209でエラーを訂正した後、前述のステップ204に進んで、訂正後のヘッダを用いてエントリを作成する。訂正不能エラーの場合は、図5のシーケンスで訂正不能エラーが検出されたときのステップ110及び111と同様のステップ210及び211を実行した後、ステップ204に進む。このとき作成されるエントリは消去されたブロックのものであるから、前述のようにOWは0にセットされる。

【0034】最後の、すなわちn番目のブロックのための処理が終了すると、ステップ202～207のループからステップ208に出る。ステップ208では、作成したばかりのメモリ管理表をアクセスすることにより、プロ

ックIDが同じでRCが異なっているブロックを探す。ステップ212でそのようなブロックが見つかった場合、それは図5のステップ108で行われるはずであった古いブロックの消去がパワーオフのために完了していないことを示す。従ってステップ213に進んで、RC

が小さい方のブロックを消去し、そのECを1だけ増分して消去したブロックに書き込み、メモリ管理表の対応するエントリにおいてECを更新すると共にOWを0にセットする。

【0035】ステップ212でブロックが見つからなかった場合、又はステップ213が終了すると、制御装置30は図6のシーケンスから出て、CPU10からのコマンドを待つ。前にも述べたように、この待機中に図5のステップ106を実行してもよい。

【0036】上述の図5及び図6のシーケンスでは説明しなかったが、よく知られているように、ブロックの書き込み及び消去後にはいわゆるベリファイが行われ、その結果ブロックが使用不可と判断されることがある。勿論、そのようなブロックがあれば、メモリ管理表のエントリを作成する必要はない。また、既に作成済みのエントリに対応するブロックが使用不可と判断された場合は、そのエントリを無効にしておけばよい。エントリの無効化は、例えばRBAを無効な値に設定しておいたり、有効無効を示すビットを設けてそれを無効状態にセットすることにより達成できる。

【0037】フラッシュ・メモリを構成するブロック（セクタ）の数、従ってメモリ管理表のエントリの数が多くなって、読取り又は書き込み時にメモリ管理表をサーチするのに時間がかかる場合は、RBAの構成要素であるヘッド又はシリンドラを単位としてメモリ管理表を分割すれば、サーチ時間を短縮できる。その場合、ブロックの消去／書き込み回数の平均化は分割した単位で、例えばシリンドラ別に行われることになり、従ってブロックの消去／書き込み回数に片寄りが生じる可能性がある。しかし、実際にそのような片寄りが生じた場合には、消去／書き込み回数の多くなったシリンドラの使用ブロックを、消去／書き込み回数の少ないシリンドラの使用ブロック又は予備ブロックと交換するようにすれば、フラッシュ・メモリ全体での平均化を図ることができる。

【0038】これまでの説明はセクタ消去型のフラッシュ・メモリを仮定していたが、複数のセクタを含むクラスタが一括消去されるクラスタ消去型のフラッシュ・メモリにも本発明を適用できる。その場合、制御装置30の動作は、図5のステップ108及び110並びに図6

のステップ210及び213でブロック消去が行われないことを除くと、基本的に図5及び図6と同じであるが、どのクラスタをいつ消去するかを決めるステップが必要になる。これは、例えば各クラスタにおいて消去すべきであると決定されたブロックの数をカウントし、それが所定の閾値に達したクラスタを消去するというステップでもよい。但しこのステップをとるときは、クラスタを消去する前に、その中に有効なブロックがもし残っていれば、その内容を他のクラスタの書き込み可能なブロック位置に移す必要がある。この有効ブロックの移動は実質的に書き込みコマンドの実行と同じであるから、制御装置30は、有効ブロックの内容をRAM32へ読み出した後、図5のステップ106及び107と同様のステップを実行することにより、有効ブロックの内容をフラッシュ・メモリ34に書き戻す。有効ブロックの移動及びクラスタの消去は、CPU10からのコマンドがないときに実行される。チップ消去型の場合も同様である。

【0039】

【発明の効果】以上のように、本発明に従えば、フラッシュ・メモリの消去及び書き込みが特定のブロックに片寄るのを防止することができ、更に消去又は書き込み中に異常パワーオフがあっても、その影響を最小限に押えることができる。

【図面の簡単な説明】

【図1】本発明に従うフラッシュ・メモリを半導体ディスク装置として使用するコンピュータ・システムの一例を示すブロック図。

【図2】半導体ディスク装置の概略構成を示すブロック図。

【図3】フラッシュ・メモリの各ブロック（セクタ）の構成を示す図。

【図4】メモリ管理表の構成を示す図。

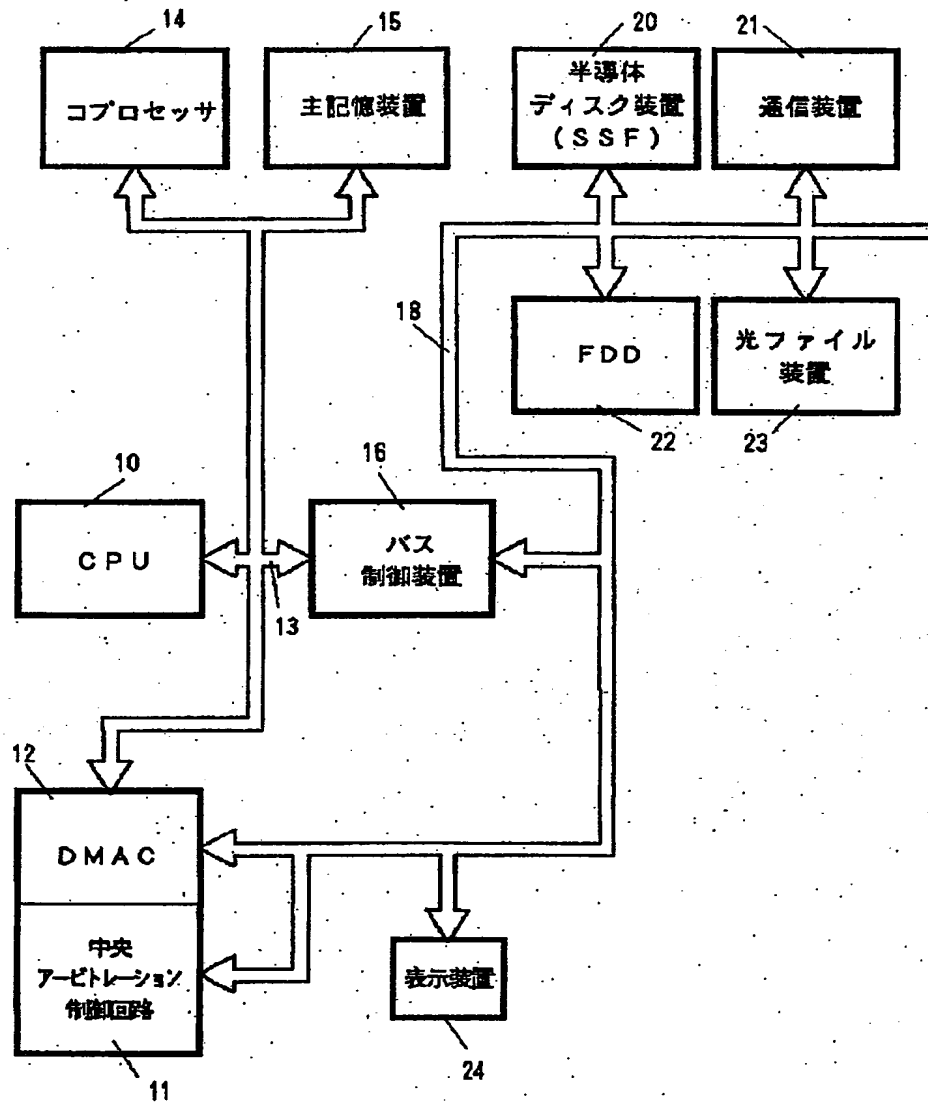
【図5】CPU10からコマンドを受け取ったときの制御装置30の動作を示すフローチャート。

【図6】パワーオン時にメモリ管理表を作成するときの制御装置30の動作を示すフローチャート。

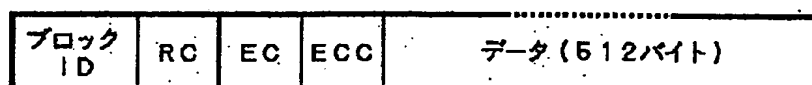
【符号の説明】

- 10 CPU
- 20 半導体ディスク装置（SSF）
- 30 制御装置
- 32 ランダム・アクセス・メモリ（RAM）
- 34 フラッシュ・メモリ
- 35 メモリ管理表

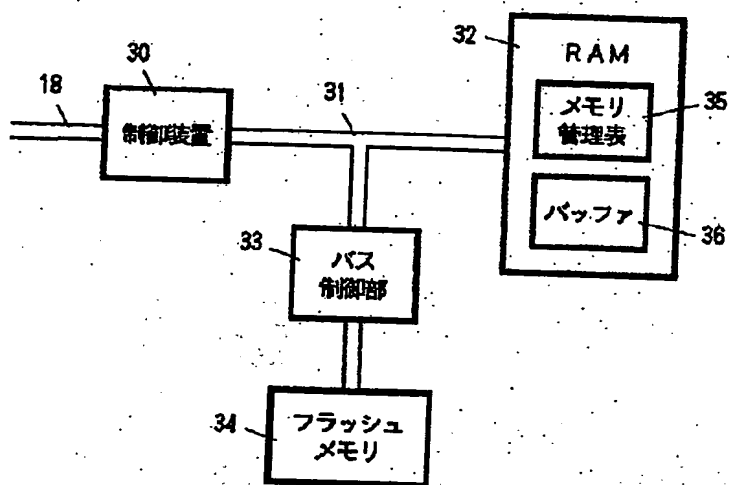
【図1】



【図3】



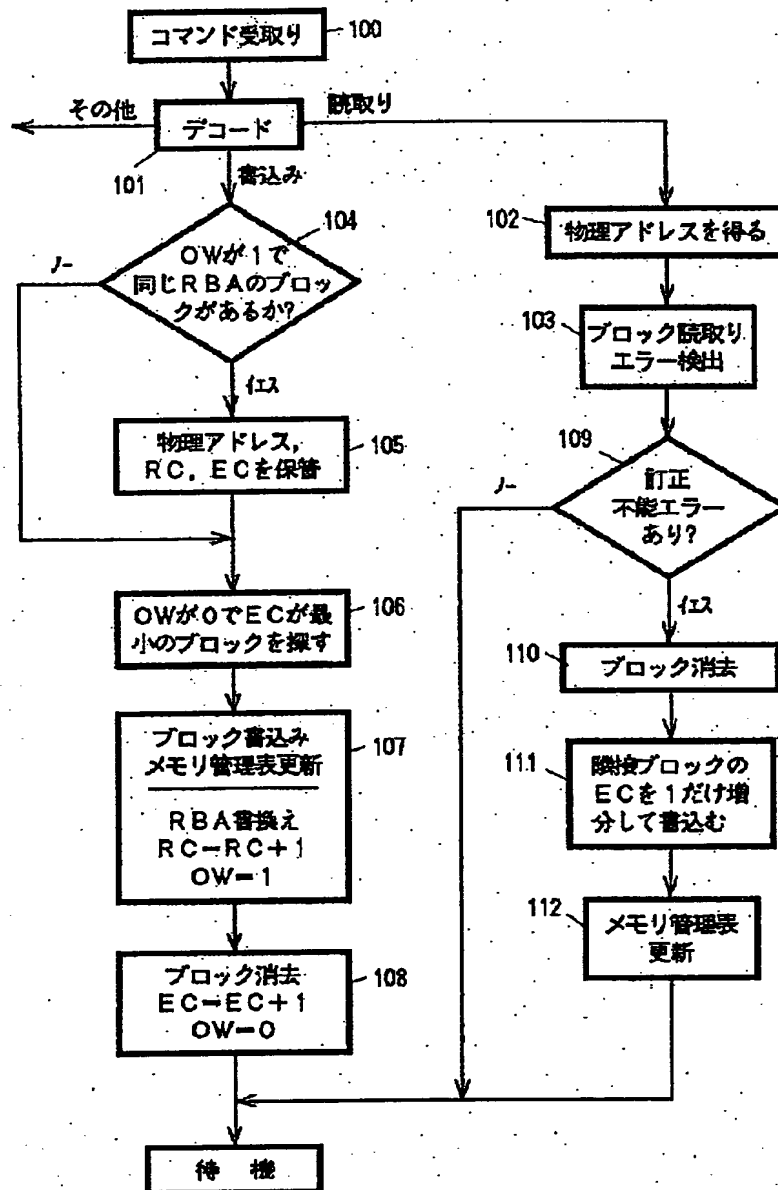
【図2】



【図4】

物理 アドレス	R B A			RC	EC	OW
	ヘッド番号	シリンダ番号	セクタ番号			
エントリ 0						
エントリ 1						
エントリ 2						
...						

【図5】



【図6】

